

*Е. П. ПАВЛЕНКО, В. М. БУТЕНКО, В. А. ГУБИН***ИССЛЕДОВАНИЕ МЕТОДОВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНОЙ ИНЖЕНЕРИИ НА ОСНОВЕ ТИПОВЫХ ПРОГРАММНЫХ ЭЛЕМЕНТОВ**

В работе рассматриваются проблемы повышения эффективности разработки ИС, и, в частности, вопросы уменьшения сроков разработки программного комплекса ИС. Проведен анализ технологий разработки программного комплекса в жизненном цикле ИС. Структурный подход к программированию предлагал осуществлять декомпозицию программ методом пошаговой детализации. Разработка структур программы выполняется при помощи построения структур входных и выходных данных, идентификации связей обработки между этими данными, формирования структуры программы на основании структур данных и обнаруженных соответствий. Преодолеть фактор сложности можно, если отойти от прямолинейного подхода к решению поставленной задачи, состоящего в последовательном и линейном наращивании исходного текста программы оператор за оператором, в результате чего получается одна длинная и аморфная программа. Оказывается действенным принцип модульности: исходная задача разбивается на относительно независимые части; они реализуются отдельными программными модулями, которые затем связываются в единое целое на этапе компоновки. Выделены особенности технологии автоматизированного синтеза программ, а именно, технологии сборки программ из типовых программных элементов. Определены основные понятия указанной выше технологии, проведено исследование конструирования программ из блоков и выявлена проблемная область. Исходными данными для постановки и решения задач синтеза системы программных модулей являются множество информационных массивов системы, для которых определены: входные, выходные и промежуточные данные; множество альтернативных процедур обработки данных; последовательности выполнения процедур в процессе обработки; способы обмена с внешней памятью. Рассмотрен подход для выделения типовых программных элементов, удовлетворяющих определенным критериям. На основании рассмотренного подхода, а также с учетом его недостатков, был предложен усовершенствованный метод классификации типовых программных элементов и способ проектирования ПО на их основе с учетом минимизации времени и стоимости проекта.

**Ключевые слова:** программное обеспечение, компьютерная инженерия, информационные системы, компоненты, типовые проектные решения, затраты на разработку.

*Є. П. ПАВЛЕНКО, В. М. БУТЕНКО, В. А. ГУБІН***ДОСЛІДЖЕННЯ МЕТОДІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ НА ОСНОВІ ТИПОВИХ ПРОГРАМНИХ ЕЛЕМЕНТІВ**

У роботі розглядаються проблеми підвищення ефективності розробки ІС, зокрема, питання зменшення термінів розробки програмного комплексу ІС. Проведено аналіз технологій розробки програмного комплексу в життєвому циклі ІС. Структурний підхід до програмування пропонував здійснювати декомпозицію програм методом покрокової деталізації. Розробка структур програми виконується за допомогою побудови структур вхідних і вихідних даних, ідентифікації зв'язків обробки між цими даними, формування структури програми на підставі структур даних і виявлених відповідей. Подолати фактор складності можна, якщо відійти від прямолінійного підходу до вирішення поставленого завдання: послідовному і лінійному нарощуванню вихідного тексту програми оператор за оператором, в результаті чого виходить одна довга і аморфна програма. Виявляється дієвим принцип модульності: вихідна задача розбивається на відносно незалежні частини; вони реалізуються окремими програмними модулями, які потім зв'язуються в єдине ціле на етапі компоновки. Виділено особливості технології автоматизованого синтезу програм, а саме, технології складання програм з типових програмних елементів. Визначено основні поняття зазначеної вище технології, проведено дослідження конструювання програм з блоків і виявлена проблемна область. Вихідними даними для постановки і рішення задач синтезу системи програмних модулів є множина інформаційних масивів системи, для яких визначені: вхідні, вихідні та проміжні дані; множина альтернативних процедур обробки даних; послідовності виконання процедур в процесі обробки; способи обміну із зовнішньою пам'яттю. Розглянуто підхід для виділення типових програмних елементів, які відповідають певним критеріям. На підставі розглянутого підходу, а також з урахуванням його недоліків, був запропонований вдосконалений метод класифікації типових програмних елементів і спосіб проектування ПЗ на їх основі з урахуванням мінімізації часу і вартості проекту.

**Ключові слова:** програмне забезпечення, комп'ютерна інженерія, інформаційні системи, компоненти, типові проектні рішення, витрати на розробку.

*Y. P. PAVLENKO, V. M. BUTENKO, V. O. GUBIN***RESEARCH OF METHODS OF DEVELOPMENT OF SOFTWARE COMPUTER ENGINEERING BASED ON TYPICAL SOFTWARE ELEMENTS**

The paper deals with the problems of increasing the effectiveness of the development of IS, and, in particular, the issues of reducing the development time of the software package of IS. The analysis of technology development software in the life cycle of IS. A structural programming approach suggested decomposing programs in a step-by-step manner. The development of program structures is carried out using the construction of input and output data structures, identification of processing links between these data, formation of a program structure based on data structures and detected matches. It is possible to overcome the complexity factor if we deviate from a straightforward approach to solving the problem posed, consisting in sequential and linear extension of the source code of the program operator-by-operator, resulting in one long and amorphous program. Here, the modularity principle is effective: the initial problem is divided into relatively independent parts; they are implemented by separate software modules, which are then linked into a single unit at the layout stage. The features of the technology of automated program synthesis, namely, the technology of assembling programs from typical program elements, are highlighted. The basic concepts of the above technology have been identified, a study has been conducted to design programs from blocks and a problem area has been identified. The initial data for the formulation and solution of problems for the synthesis of a system of program modules are the set of information arrays of the system, for which there are defined: input, output and intermediate data; many alternative data processing procedures; sequence of procedures in the processing; ways of sharing with external memory. The approach to the selection of typical program elements that meet certain criteria is considered. On the basis of the considered approach, and also taking into account its shortcomings, an improved method was proposed for classifying typical program elements and a method for designing software based on them, taking into account minimizing the time and cost of the project.

**Keywords:** software, computer engineering, information systems, components, typical design solutions, development costs.

© Е. П. Павленко, В. М. Бутенко, В. А. Губин, 2019

**Введение.** Создание программных систем для автоматизации предприятий является сложной и поэтапной задачей, поэтому для определения необходимых ресурсов, в том числе времени, важно регламентировать порядок разработки и использования некоторых видов типовых программных элементов по стадиям проектирования системы.

Для ускорения времени разработки и повышения качества кода программы разработано большое количество методик, технологий и подходов. Все они относятся не только к организации жизненного цикла ПО, но и к построению рациональной структуры организации, занимающейся разработкой информационных систем (ИС), и к каждому разработчику в частности.

Одним из решений проблем создания программного комплекса компьютерной инженерии является использование технологий автоматизации программирования, которые предполагают сокращение сроков разработки программных продуктов и повышение их характеристик качества.

Недостатки, изначально заложенные в требованиях на систему, невозможно компенсировать последующими версиями программных средств на различных этапах жизненного цикла. Поэтому проблема создания рациональной структуры ПО ИС является ключевой – её решение открывает новые возможности повышения эффективности ИС, а практическая значимость возрастает по мере усложнения и интеграции задач в ИС.

**Постановка проблемы.** Таким образом, можно прийти к выводу, что эффективность разработки ПО ИС определяется совокупностью множества факторов. Анализ показал многообразие существующих подходов к разработке ПО, их достоинства и недостатки, а также выявил общие задачи этапа реализации ПО ИС. Проанализированы структурный и объектно-ориентированный подходы к разработке ПО ИС.

Существенную помощь на этапе проектирования могут оказать разнообразные CASE-средства, но с их помощью можно построить лишь костяк ПО, их использование подразумевает дальнейшую доработку бизнес логики и отладку всего ПО.

На стадии проектирования ИС модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм.

Таким образом, возникает задача исследования технологий использования синтеза программ из типовых элементов, позволяющих избежать доработки выходного продукта, а также в исследовании повышения эффективности их применения.

**Анализ состояния проблемы.** Структурный подход к программированию в том виде, в котором он был сформулирован в 70-х годах XX в., предлагал осуществлять декомпозицию программ методом пошаговой детализации [1]. Результатом декомпозиции является структурная схема программы, которая представляет собой многоуровневую иерархическую схему взаимодействия подпрограмм по управлению. Минимально такая схема отображает два уровня иерархии, т. е. показывает общую структуру программы [2].

Однако тот же метод позволяет получить структурные схемы с большим количеством уровней.

Для анализа технологичности полученной иерархии модулей используют структурные карты Константа [3] или Джексона [4]. На структурной карте отношения между модулями представляют в виде графа, вершинам которого соответствуют модули и общие области данных, а дугам – межмодульные вызовы и обращения к общим областям данных.

Существуют методики проектирования программного обеспечения Джексона [5] и Варнье – Орра [6], основанные на декомпозиции данных. При создании своей методики авторы исходили из того, что структуры исходных данных и результатов определяют структуру программы.

Разработка структур программы в соответствии с [5] выполняется следующим образом: строят изображение структур входных и выходных данных; выполняют идентификацию связей обработки между этими данными; формируют структуру программы на основании структур данных и обнаруженных соответствий; добавляют блоки обработки элементов, для которых не обнаружены соответствия; анализируют и обрабатывают несоответствия; добавляют необходимые операции (ввод, вывод, открытие/закрытие файлов); записывают программу в структурной нотации.

Однако методика [5, 6] может использоваться только в том случае, если данные разрабатываемых программ могут быть представлены в виде иерархии или совокупности иерархий.

Практика создания сложных программных систем говорит о том, что преодолеть фактор сложности можно, если отойти от прямолинейного подхода к решению поставленной задачи, состоящего в последовательном и линейном наращивании исходного текста программы оператор за оператором, строка за строкой, в результате чего получается одна длинная и аморфная программа. Здесь оказывается действенным принцип модульности. Он заключается в том, что исходная задача разбивается на относительно независимые части; они реализуются отдельными программными модулями, которые затем связываются в единое целое на этапе компоновки [7].

Процесс разбиения одной сложной задачи на подзадачи и распределение функций между модулями определяется принятым методом проектирования, а синтаксические и функциональные средства оформления программных модулей в виде законченных программных единиц определяются возможностями алгоритмического языка и операционной системы.

Для объектно-ориентированных программных систем применяется иной подход [8]. Поскольку экземпляры классов обмениваются сообщениями, определяют для каждого класса поступающие его экземплярам сообщения и на их основе строят диаграммы перехода (описать класс как конечный автомат). Строят также модели состояний для каждого объекта и определяют списки событий, изменяющих состояние объектов.

После выделения классов и их неформального описания строятся модели процессов, которые должны

быть реализованы в будущем ПО. В такой модели отражаются внешние события (действия пользователя) и вызываемые этими событиями действия с экземплярами классов. Если пользователь указал все характеристики данного правильно, нужно построить экземпляр объекта “данное” и включить его в множество данных [9].

Разработка объектно-ориентированных программных систем для железнодорожного транспорта имеет свою специфику, в частности, необходимость поддерживать многоуровневые структуры данных и реализовывать структуру классов с наследованием [10], [11]. Выбор технологии разработки ПО для таких систем предлагается выполнять с помощью процедур, описанных в [12]. Методы разработки программного обеспечения возможно использовать в распределенных вычислениях в различных компонентах информационных систем на железнодорожном транспорте [13].

Таким образом, налицо многообразие существующих подходов к разработке ПО, имеющих свои достоинства и недостатки. Выявились также общие задачи этапа реализации ПО ИС.

**Метод решения проблемы.** Причины применения технологии типового проектирования ПО:

- существенно снижаются затраты на проектирование, разработку и модернизацию ИС;
- больше возможностей обеспечивать должный научно-технический уровень разработки ИС (в отличие от технологии индивидуального проектирования).

Технология типового проектирования ПО является одной из разновидностей индустриального проектирования. Заключается в создании ПО информационной системы из готовых типовых элементов.

Область применения: автоматизация деятельности таких объектов, для которых характерны общие правила функционирования и управления. В первую очередь, сюда относятся экономические системы, для которых характерны:

- схожая структура и правила управления;
- единые стандарты отчетности;
- схожие комплексы используемых технических и программных средств;
- единая цель существования: извлечение прибыли.

Процесс проектирования ПО ИС состоит из следующих основных этапов:

- разбиение проекта информационной системы на отдельные составляющие (компоненты);
- выбор и приобретения имеющихся на рынке типовых проектных решений (тиражируемых продуктов) для каждого компонента ИС;
- настройка и доработка приобретенных типовых проектных решений в соответствии с требованиями конкретной предметной области.

Должна быть принципиальная возможность декомпозиции информационной системы на множество составляющих компонентов (комплексов, подсистем, программных модулей). В зависимости от уровня декомпозиции информационной системы различают следующие классы типовых проектных решений: элементные, подсистемные, объектные.

Типовое проектное решение (ТПР) – это представленное в виде комплекта проектной документации или набора программных модулей проектное решение, пригодное к многократному использованию.

Основные черты ТПР:

- типовые проектные решения ориентированы на автоматизацию деятельности множества однородных объектов (путем настройки под конкретные особенности каждого из них);
- основная цель применения ТПР – уменьшение трудоемкости и стоимости проектирования или разработки ПО ИС;
- создание ТПР возможно только после тщательного и всестороннего изучения предметной области и предполагает обобщение накопленного в частных случаях опыта (путем классификации, типизации, абстрагирования, унификации).

Типовые решения бывают простыми или комбинированными. Простые ТПР охватывают только какой-либо один вид обеспечения ИС, комбинированные – два и более. Примеры простых ТПР: классификаторы ИО, прикладные программы общего и специального назначения, инструктирующие руководства по управлению бизнес-процессами, рекомендации по составлению ТЗ.

Требования, выдвигаемые к типовым проектным решениям:

- возможность использования для создания новой ИС при минимальном участии разработчиков ТПР;
- соответствие требованиям положений и стандартов, распространяемых на информационную систему в целом или ее часть;
- способность удовлетворять максимально возможному числу потребностей в рамках своего функционального назначения;
- возможность адаптации к конкретным условиям проекта путем изменения параметров.

В качестве типового элемента при элементном проектировании используются простые ТПР, относящиеся к отдельной задаче ИС. В этом случае ИС комплектуется как множество ТПР по отдельным разрозненным задачам. Дополнительные элементы, для которых отсутствуют ТПР, разрабатываются вручную.

Исходными данными для постановки и решения динамических задач синтеза системы модулей являются множество информационных массивов системы, для которых определены: входные, выходные и промежуточные данные; множество альтернативных процедур обработки данных, входящих в состав алгоритмов, обеспечивающих решение задач системы; возможные последовательности выполнения процедур в процессе обработки; способы обмена с внешней памятью модульной системы обработки данных; взаимосвязи процедур с информационными массивами; временные характеристики обращения к массивам.

Выделим множество программных блоков

$$P = \{P_1, P_2, \dots, P_n\},$$

которые можно использовать при проектировании ПО ИС. Каждый блок характеризуется затратами на

разработку  $Z_1, Z_2, \dots, Z_n$  и временем на разработку  $T_1, T_2, \dots, T_n$ .

Функциональность программной системы включает в себя  $m$  различных функций по обработке информации

$$F = \{f_1, f_2, \dots, f_m\}.$$

Можно выделить два основных критерия оценки эффективности синтеза ПО ИС из функциональных блоков: затраты и время.

Постановка задачи формулируется следующим образом: найти такое подмножество программных блоков

$$\begin{aligned} P' &\subset P, \\ P' &= \{P_i, P_k, \dots, P_l\}, \end{aligned}$$

чтобы реализовать все  $m$  функций по обработке информации.

Ограничения:

$$\begin{aligned} Z &= Z_{\text{сопр}} + Z_i + Z_k + \dots + Z_l \rightarrow \min, \\ T &= T_{\text{сопр}} + T_i + T_k + \dots + T_l \rightarrow \min, \end{aligned}$$

где  $Z_{\text{сопр}}$  – затраты на сопряжение программных блоков,

$T_{\text{сопр}}$  – время на сопряжение программных блоков.

Предлагается следующий метод решения задачи выбора типовых программных элементов.

Шаг 1. Рассмотрим функцию  $f_1$ . Если программный блок  $P_1$  необходим при реализации этой функции, то  $Z = Z_1$ ,  $T = T_1$ .

Шаг 2. Если программный блок  $P_i$  необходим при реализации этой функции, то  $Z = Z + Z_i$ ,  $T = T + T_i$ . Иначе переходим к шагу 3.

Шаг 3. Если функция  $f_1$  реализована полностью, то переходим к рассмотрению функции  $f_2$ .

Шаг 4. Если программный блок  $P_1$  необходим при реализации этой функции и он не был задействован при реализации предыдущей функции, то  $Z = Z + Z_1$ ,  $T = T + T_1$ .

Шаг 5. Если программный блок  $P_i$  необходим при реализации функции, то  $Z = Z + Z_i$ ,  $T = T + T_i$ . Иначе переходим к шагу 5.

Шаг 6. Если функция реализована полностью, то переходим к рассмотрению следующей функции

Шаг 7. Если все функции рассмотрены, то оцениваем  $Z_{\text{сопр}}$  и  $T_{\text{сопр}}$  и фиксируем затраты и время по первому варианту рассмотрения.

$$\begin{aligned} Z^{(1)} &= Z_{\text{сопр}} + Z, \\ T^{(1)} &= T_{\text{сопр}} + T. \end{aligned}$$

Шаг 8. Начнем с рассмотрения функции  $f_2$  и повторим шаги 1–7. Зафиксируем затраты и время по следующему варианту рассмотрения.

Шаг 9. Определим

$$Z = \min_i Z^{(i)}.$$

Таким образом, мы определили пару  $(Z, T)$ , а также подмножество программных блоков  $P'$ , являющееся решением задачи.

**Выводы.** В работе проведён анализ проблемы обоснования выбора программных компонентов при

проектировании программного обеспечения ИС, анализ технологий разработки программного комплекса в жизненном цикле ИС, исследован принцип модульности при проектировании ПО.

На основании проведённого исследования разработан метод решения задачи выбора типовых программных элементов, предложены критерии выбора программных элементов, разработан алгоритм решения задачи обоснования выбора программных компонентов. Полученные результаты работы опробованы на реальном примере – при проектировании ПО ИС, состоящей из 4 функциональных задач.

Разработанный метод в результате небольших изменений может быть использован для обоснования выбора программных элементов при проектировании других видов ПО, и разрабатываемое программное обеспечение может быть улучшено за счёт увеличения его функциональности и сокращения затрат на разработку.

#### Список литературы

1. Структурное программирование. URL: <http://www.maksakov-sa.ru/TehProgram/StrProgr/index.html> (дата обращения: 16.12.2018)
2. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. URL: <https://www.livelib.ru/book/1001405237/about-teoriya-i-praktika-strukturnogo-programmirovaniya-richard-linger> (дата обращения: 16.12.2018)
3. Средства структурного проектирования. URL: <http://www.interface.ru/fset.asp?Url=/case/defs71.htm> (дата обращения: 04.02.2019)
4. Структурные карты Джексона. URL: <http://mash-xxl.info/article/433911/> (дата обращения: 04.02.2019)
5. Проектирование программного обеспечения. Основы программной инженерии. URL: <http://swebok.sorlik.ru/2-software-engineering-designi> (дата обращения: 04.02.2019)
6. Современные методологии структурного анализа и проектирования. URL: <http://swsys.ru/index.php?page=article&id=1025> (дата обращения: 04.02.2019)
7. Проектирование программного обеспечения. URL: <http://window.edu.ru/library/pdf2txt/965/28965> (дата обращения: 04.02.2019)
8. Объектно-ориентированный анализ и проектирование. URL: <http://www.hardline.ru/1/5/1390/1789-6.htm> (дата обращения: 04.02.2019)
9. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. URL: <https://www.twirpx.com/file/279137> (дата обращения: 04.02.2019)
10. Listrovoy S. V., Butenko V. M., Bryksin V. O., Golovko O. V. Development of method of definition maximum clique in a non-oriented graph. *Eastern European Journal of Enterprise Technologies*. 2017. Vol. 5, № 4 (89). – P. 12–17. EID: 2-s2.0-85032585697 DOI: 10.15587/1729-4061.2017.111056).
11. Лістровий С. В., Панченко С. В., Мойсєєнко В. І., Бутєнко В. М. Математичне моделювання в розподілених інформаційно-керуючих системах залізничного транспорту: Монографія. Харків: ФОП Бровін О. В., 2017. 220 с
12. Павленко Е. П., Берюх И. С., Айвазов В. А. Выбор технологии разработки программного обеспечения информационных систем. *ScienceRise*. Харьков, 2015. № 5/2. С. 40–43
13. S. V. Listrovoy, O. V. Golovko, V. M. Butenko, M. V. Ushakov Formulation of the Problem of Maximum Clique Determination in Non-Oriented Graphs. *International Journal of Engineering & Technology*. Vol. 7, no. 4.3 (2018): Special Issue 3. P. 293–297.

#### References (transliterated)

1. *Strukturnoye programmirovaniye*. [Structured programming] Available at: <http://www.maksakov-sa.ru/TehProgram/StrProgr/index.html> (accessed 16.12.2018).

2. Linger R., Mills H., Witt B. *Teoriya i praktika strukturnogo programmirovaniya*. [Theory and practice of structured programming] Available at: <https://www.livelib.ru/book/1001405237/about-teoriya-i-praktika-strukturnogo-programmirovaniya-richard-linger> (accessed 16.12.2018).
3. *Sredstva strukturnogo proyektirovaniya*. [Structural Design Tools] Available at: <http://www.interface.ru/fset.asp?Url=/case/defs71.htm> (accessed 04.02.2019).
4. *Strukturnyye karty Dzheksona*. [Jackson Structural Maps] Available at: <http://mash-xxl.info/article/433911> (accessed 04.02.2019).
5. *Proyektirovaniye programmnoy obespecheniya. Osnovy programmnoy inzhenerii*. [Software design. Software Engineering Basics] Available at: [http://swebok.sorlik.ru/2-software\\_engineering\\_designi](http://swebok.sorlik.ru/2-software_engineering_designi) (accessed 04.02.2019).
6. *Sovremennyye metodologii strukturnogo analiza i proyektirovaniya*. [Modern methodologies of structural analysis and design] Available at: <http://swsys.ru/index.php?page=article&id=1025> (accessed 04.02.2019).
7. *Proyektirovaniye programmnoy obespecheniya*. [Software design] Available at: <http://window.edu.ru/library/pdf2txt/965/28965> (accessed 04.02.2019).
8. *Ob'yektno-oriyentirovanny analiz i proyektirovaniye*. [Object-oriented analysis and design] Available at: <http://www.hardline.ru/1/5/1390/1789-6.htm> (accessed 04.02.2019).
9. Booch G. *Ob'yektno-oriyentirovanny analiz i proyektirovaniye s primerami prilozheniy*. [Object-oriented analysis and design with examples of applications] Available at: <https://www.twirpx.com/file/279137> (accessed 04.02.2019).
10. Listrovoy S. V., Butenko V. M., Bryksin V. O., Golovko O. V. Development of method of definition maximum clique in a non-oriented graph. *Eastern European Journal of Enterprise Technologies*. 2017, vol. 5, no. 4 (89), pp. 12–17. EID: 2-s2.0-85032585697 DOI: 10.15587/1729-4061.2017.111056).
11. Listrovoy S. V., Panchenko S. V., Mojseenko V. I., Butenko V. M. *Matematychnye modelyuvannya v rozpodilennykh informatsiynno-keruyuchykh systemakh zaliznychnoho transportu* [Mathematical modeling in distributed information-control systems of railway transport]. Kharkiv, 2017. 220 p.
12. Pavlenko E. P., Beruh I. C., Ayvazov V. A. Vybory tekhnologii razrabotki programmnoy obespecheniya informatsionnykh sistem [The choice of technology for developing software information systems]. *ScienceRise* [ScienceRise]. Kharkov, 2015, no. 5/2, pp. 40–43.
13. Listrovoy S. V., Golovko O. V., Butenko V. M., Ushakov M. V. Formulation of the Problem of Maximum Clique Determination in Non-Oriented Graphs. *International Journal of Engineering & Technology*. 2018, vol. 7, no. 4.3: Special Issue 3, pp. 293–297.

Поступила (received) 19.02.2019

#### Відомості про авторів / Сведения об авторах / About the Authors

**Павленко Євген Петрович (Павленко Евгений Петрович, Pavlenko Yevhen Petrovych)** – кандидат технічних наук, доцент, Український державний університет залізничного транспорту, доцент кафедри спеціалізованих комп'ютерних систем; м. Харків, Україна; ORCID: <http://orcid.org/0000-0002-7626-9933>; e-mail: [evgenijpavlenko821@gmail.com](mailto:evgenijpavlenko821@gmail.com)

**Бутенко Володимир Михайлович (Бутенко Владимир Михайлович, Butenko Vladimir Mihajlovych)** – кандидат технічних наук, доцент, Український державний університет залізничного транспорту, доцент кафедри спеціалізованих комп'ютерних систем; м. Харків, Україна; ORCID: <http://orcid.org/0000-0001-9958-3960>; e-mail: [docent.butenko@gmail.com](mailto:docent.butenko@gmail.com)

**Губін Вадим Олександрович (Губин Вадим Александрович, Gubin Vadim Oleksandrovyich)** – Національний університет радіоелектроніки, старший викладач кафедри штучного інтелекту; м. Харків, Україна; ORCID: <http://orcid.org/0000-0003-1850-1930>; e-mail: [vadim.gubin@nure.ua](mailto:vadim.gubin@nure.ua)